

AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A computer-implemented method of identifying compatible software threads to execute on a Simultaneous Multi-Threading (SMT) processor, said method comprising:

identifying a time interval during which both a first thread and a second thread are executing on the SMT processor;

retrieving a performance value that is a cycles per instruction (CPI) value and that occurred during the identified time interval;

determining, based upon the retrieved performance value, whether the first thread is compatible with the second thread, wherein the determining includes comparing the CPI value to a threshold value, wherein the first thread and second thread are determined to be compatible if the CPI value is better than the threshold value; and

recording the compatibility of the first thread with the second thread in response to the determination.
2. (Canceled)
3. (Previously Presented)) The method as described in claim 1 wherein the CPI value further comprises:

retrieving a number of cycles value indicating the number of processing cycles that occurred during the time interval;

retrieving a number of instructions value indicating the number of instructions that were executed by the SMT processor during the time interval; and

dividing the number of cycles value by the number of instructions value, the dividing resulting in the CPI value.

4. (Canceled)
5. (Previously Presented) The method as described in claim 1 further comprising:
writing a first identifier corresponding to the first thread and the CPI value to a compatibility list that corresponds to the second thread.
6. (Original) The method as described in claim 5 wherein the writing is performed in response to identifying an empty field in the second thread's compatibility list.
7. (Original) The method as described in claim 5 wherein the writing is performed in response to: behavior

comparing the CPI value to one or more previously recorded CPI values that correspond to one or more previously identified compatible threads; and

determining that the CPI value is better than at least one of the previously recorded CPI values.
8. (Original) The method as described in claim 7 further comprising:

removing one of the previously recorded CPI values and data corresponding to one of the previously identified compatible threads prior to the writing.
9. (Original) The method as described in claim 1 further comprising:

writing a first identifier corresponding to the first thread to a compatibility list corresponding to the second thread, wherein the compatibility list stores a plurality of thread identifiers compatible with the second thread.

10. (Original) The method as described in claim 9 further comprising:

writing a timestamp corresponding to the first identifier, the timestamp indicating a time at which the time interval occurred, wherein each of the plurality of thread identifiers also include a plurality of timestamps indicating when each of the threads executed with the second thread.

11. (Original) The method as described in claim 10 further comprising:

periodically cleaning a plurality of compatibility lists, including the second thread's compatibility list, the cleaning including:

reading the entries corresponding to each of the threads listed in the compatibility lists;

comparing the timestamps listed in the compatibility list with a current time;

determining, based on the comparison, whether the entry associated with the timestamp is stale; and

removing the entry in response to determining that it is a stale entry.

12. (Original) The method as described in claim 1 further comprising:

sensing that either the first thread or the second thread is about to complete;

scheduling a new thread to execute, the scheduling comprising:

identifying a compatible thread, the compatible thread being compatible to the thread that is not about to complete;

determining whether the compatible thread is ready to execute; and
 dispatching the compatible thread to execute on the SMT processor.

13. (Original) The method as described in claim 12 wherein the thread that is about to complete and the compatible thread are listed in a first run queue and wherein the thread that is not about to complete is listed in a second run queue.
14. (Previously Presented) A computer-implemented method of dispatching software threads to execute on a Simultaneous Multi-Threading (SMT) processor, said method comprising:
 sensing that a completing thread is about to complete execution on the SMT processor;
 identifying a running thread that is still executing on the SMT processor;
 checking a list of one or more compatible threads, wherein the compatible threads are compatible with the running thread, and wherein the compatibility is based on a comparison of a first cycles per instruction (CPI) value corresponding to the running thread with one or more CPI values corresponding to each of the listed compatible threads;
 determining that one of the compatible threads is ready to execute; and
 dispatching the determined thread to execute on the SMT processor.
15. (Original) The method as described in claim 14 wherein the completing thread and the compatible threads are listed in a first run queue and wherein the running thread is listed in a second run queue.

16. (Previously Presented) The method as described in claim 14 wherein the determination that one of the compatible threads is ready to execute further comprises:

checking whether the compatible threads are ready to execute in order of the one or more CPI values corresponding to each of the listed compatible threads, so that the compatible threads with corresponding CPI values that are closer to the running thread's CPI value are checked before the compatible threads with corresponding CPI values that are farther from the running thread's CPI value.

17. (Canceled)

18. (Canceled)

19. (Canceled)

20. (Canceled)

21. (Canceled)

22. (Canceled)

23. (Canceled)

24. (Canceled)

25. (Canceled)

26. (Canceled)

- 27. (Canceled)
- 28. (Canceled)
- 29. (Canceled)
- 30. (Canceled)
- 31. (Canceled)
- 32. (Canceled)
- 33. (Canceled)
- 34. (Canceled)
- 35. (Canceled)
- 36. (Canceled)
- 37. (Canceled)
- 38. (Canceled)
- 39. (Canceled)
- 40. (Canceled)
- 41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)